

W E L C O M E T O

F L I N T

T E C H H U B

H E L P I N G Y O U T O U N D E R S T A N D T E C H

UNDERSTANDING

PYTHON

CODING

Objectives

INTRO TO PYTHON PROGRAMMING

OBJECTIVES:

- LEARN PYTHON BASICS
- WRITE SIMPLE PROGRAMS
- UNDERSTAND CODING BEST PRACTICES

PYTHON IS ONE OF THE MOST POPULAR AND BEGINNER-FRIENDLY PROGRAMMING LANGUAGES.

Why Python?

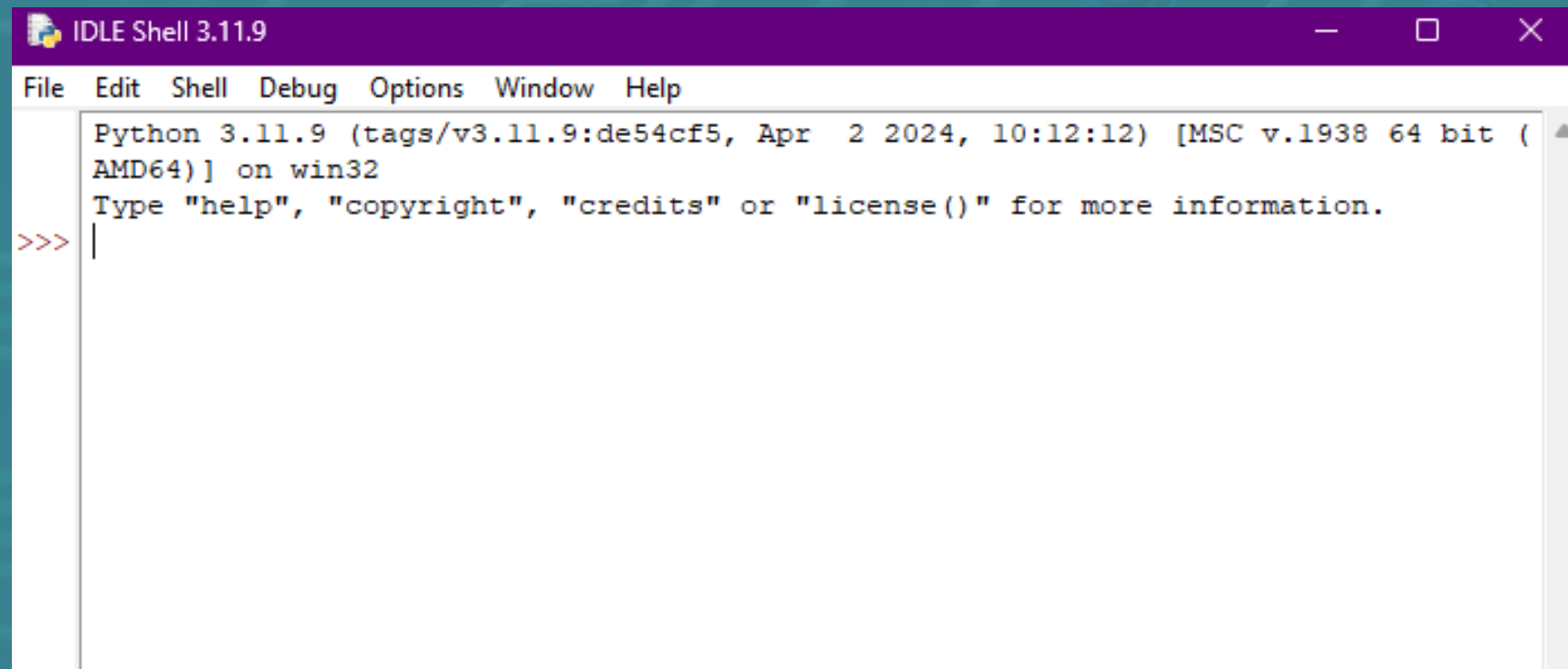
- EASY TO LEARN, READABLE SYNTAX
- WIDELY USED: WEB, DATA SCIENCE, AI,
AUTOMATION
- HUGE COMMUNITY & LIBRARIES



PYTHON IS VERSATILE — IT POWERS EVERYTHING FROM WEBSITES TO AI.

Setting up Python

- INSTALL PYTHON (PYTHON.ORG)
- USE IDEs: VS CODE, PYCHARM, JUPYTER NOTEBOOK
- RUN CODE IN TERMINAL OR EDITOR

A screenshot of the Python IDLE Shell 3.11.9 window. The window has a purple title bar with the text "IDLE Shell 3.11.9" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area is a text editor with a white background and black text. It displays the Python 3.11.9 startup message: "Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32" followed by "Type 'help', 'copyright', 'credits' or 'license()' for more information." and a prompt ">>>".

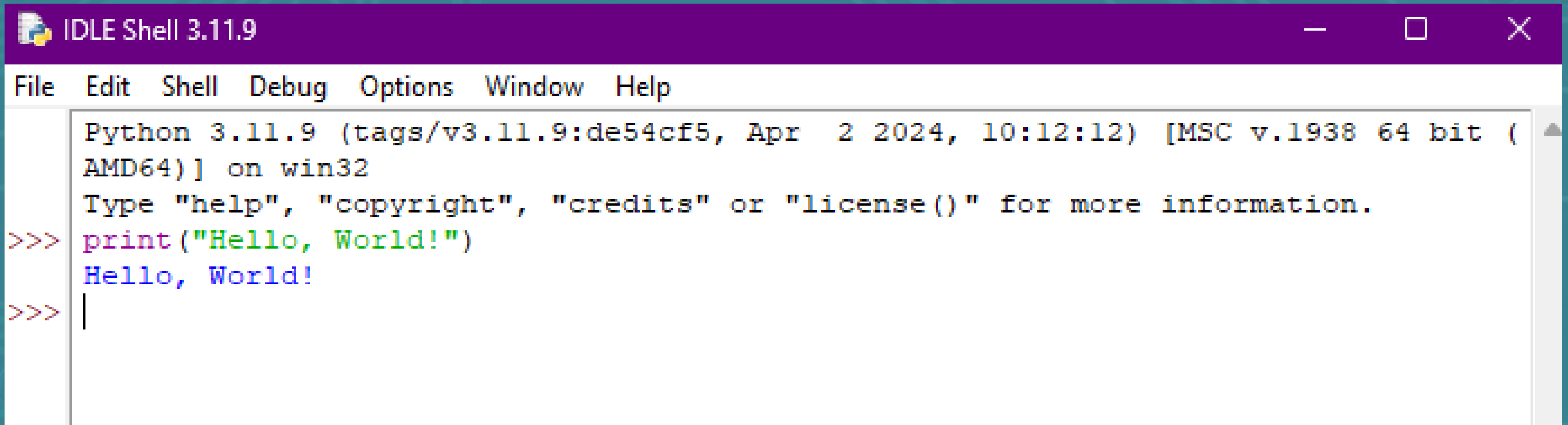
```
IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

Screenshot of the Python editor (IDE) from the python.org download

LET'S SET UP YOUR CODING ENVIRONMENT.

Your First Program

- PRINTING TEXT ON THE SCREEN, EVERY CODERS FIRST STEP!

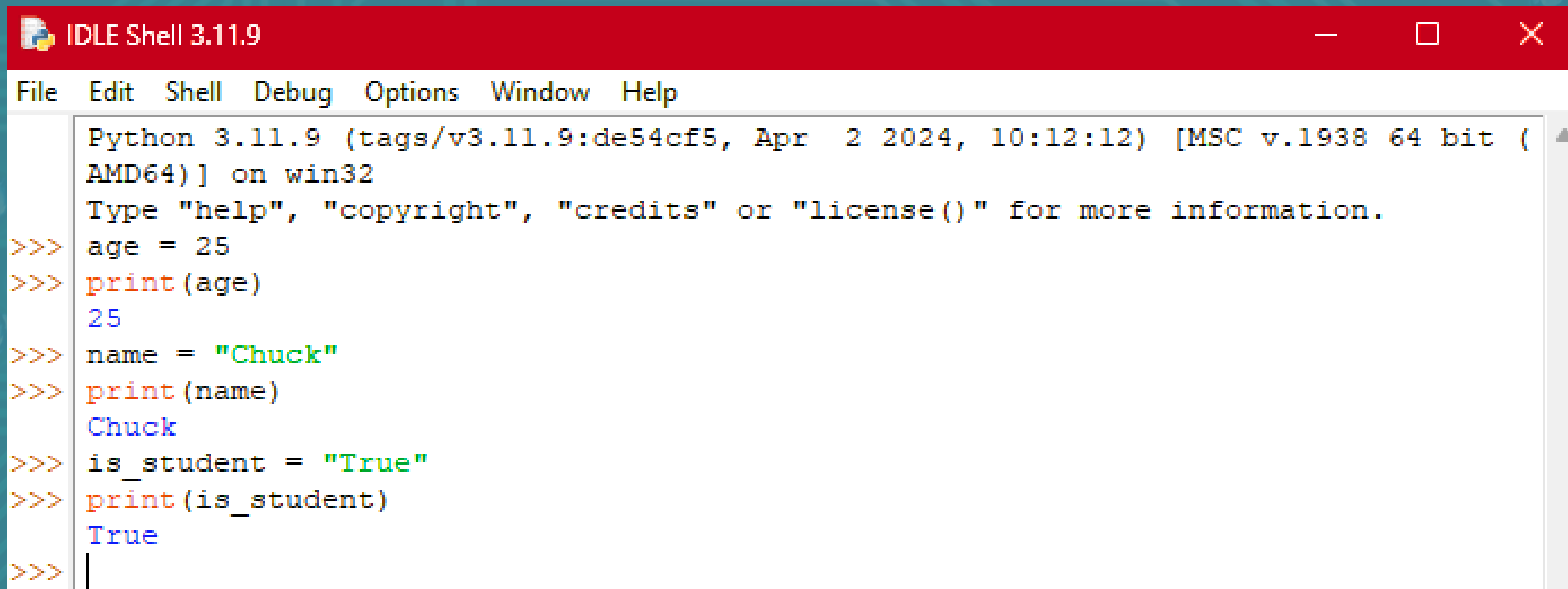
A screenshot of the IDLE Shell 3.11.9 window. The window has a purple title bar with the text 'IDLE Shell 3.11.9' and standard window controls. Below the title bar is a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the Python 3.11.9 shell prompt. The output of the program is displayed, showing the version information and the result of the print statement.

```
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello, World!")
Hello, World!
>>> |
```

THIS IS THE SIMPLEST PYTHON PROGRAM.

Variables & Data Types

- VARIABLES STORE VALUES
- COMMON TYPES: INT, FLOAT, STR, BOOL

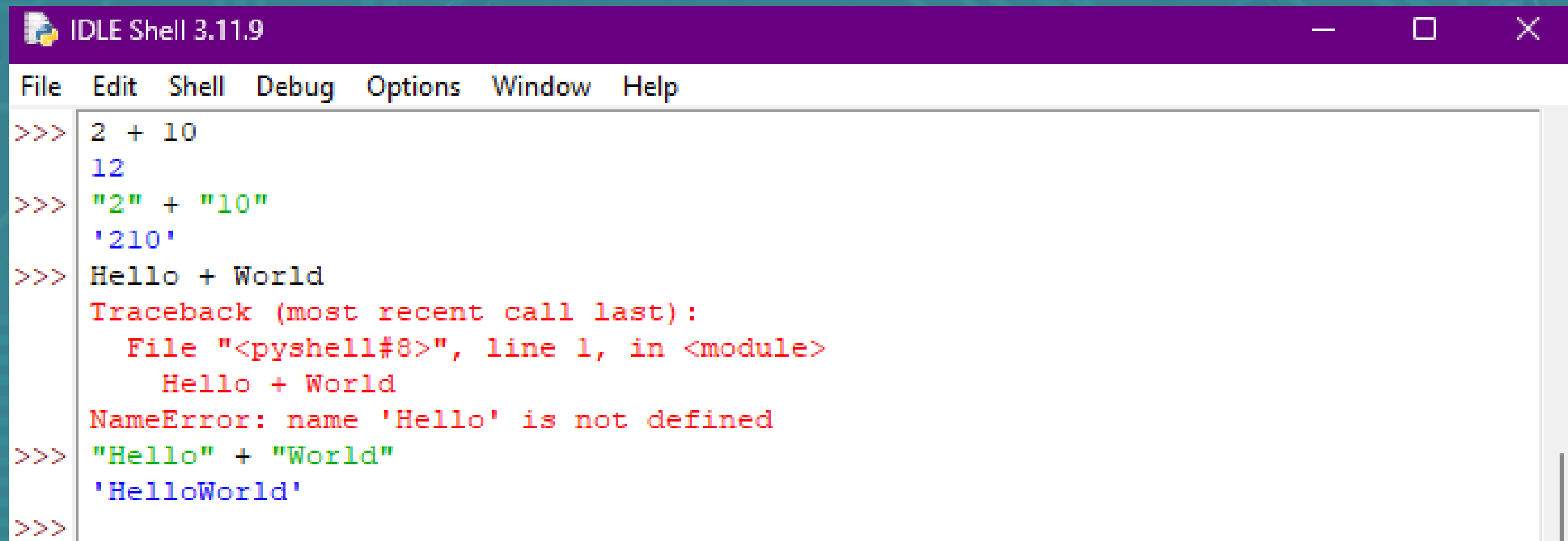


```
IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> age = 25
>>> print(age)
25
>>> name = "Chuck"
>>> print(name)
Chuck
>>> is_student = "True"
>>> print(is_student)
True
>>> |
```

VARIABLES ARE LIKE LABELED BOXES FOR DATA.

Basic Operations

- ARITHMETIC: +, -, *, /, %
- STRING CONCATENATION: "HELLO" + "WORLD"
- BOOLEAN LOGIC: AND, OR, NOT

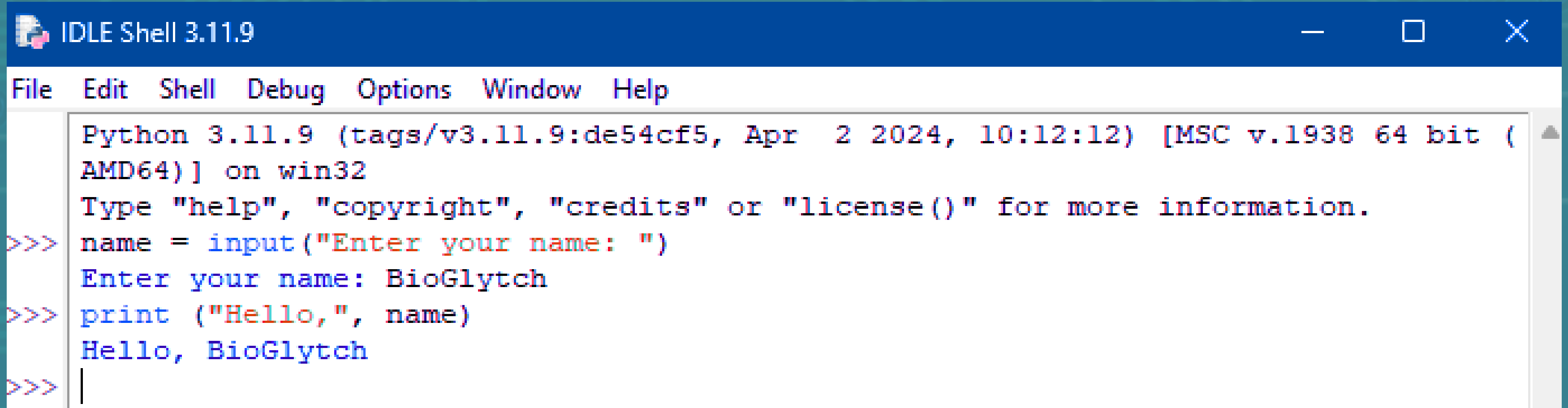
A screenshot of the IDLE Shell 3.11.9 window. The window has a purple title bar with the text "IDLE Shell 3.11.9" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a white text editor with a light blue border. It contains the following text:

```
>>> 2 + 10
12
>>> "2" + "10"
'210'
>>> Hello + World
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    Hello + World
NameError: name 'Hello' is not defined
>>> "Hello" + "World"
'HelloWorld'
>>>
```

VARIABLES ARE LIKE LABELED BOXES FOR DATA.

Input & Output

- INPUT() FOR USER INPUT
- PRINT() FOR OUTPUT

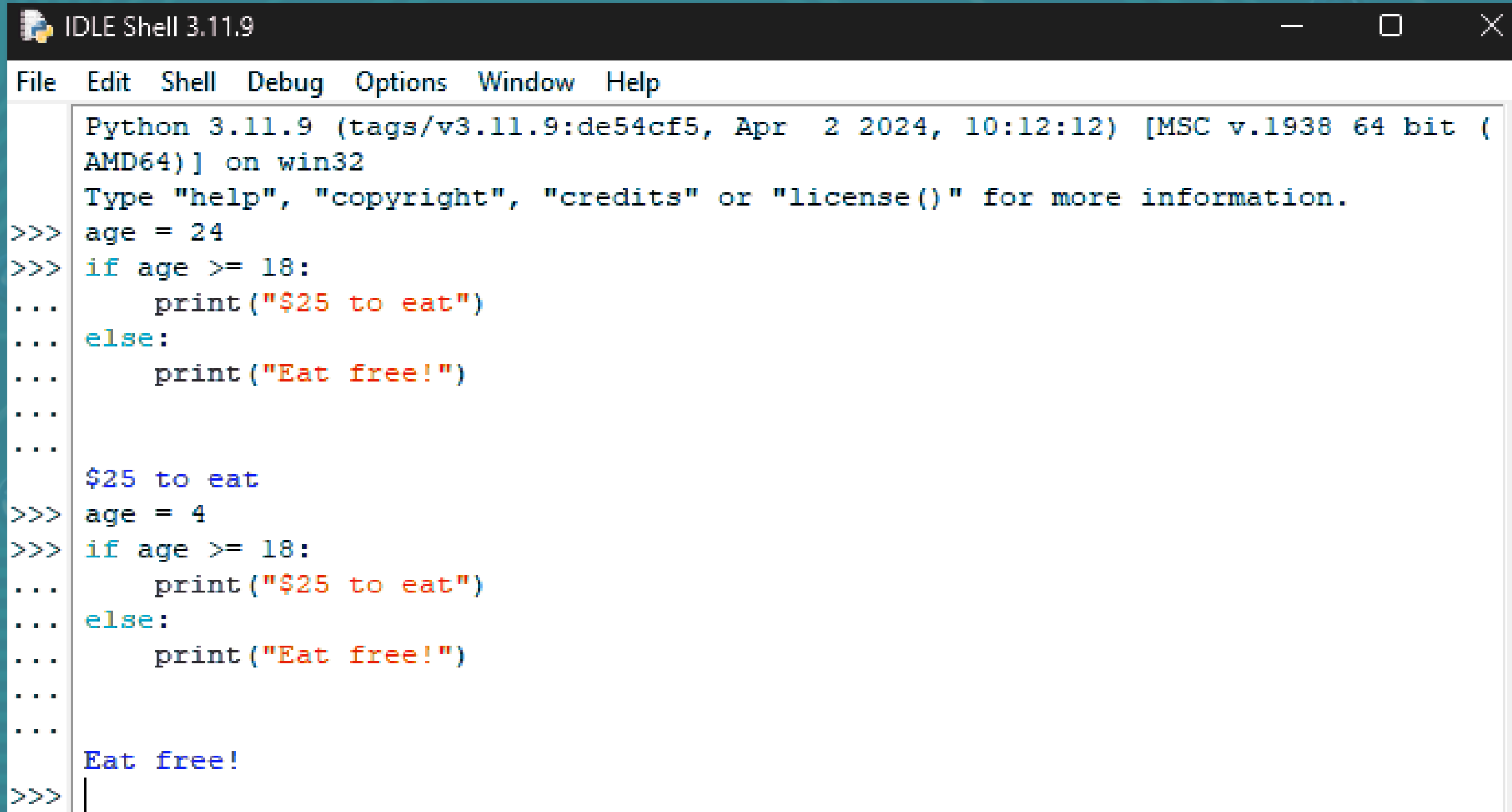


```
IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> name = input("Enter your name: ")
Enter your name: BioGlytch
>>> print ("Hello,", name)
Hello, BioGlytch
>>> |
```

PROGRAMS CAN INTERACT WITH USERS.

Conditional Statements

- IF, ELIF, ELSE

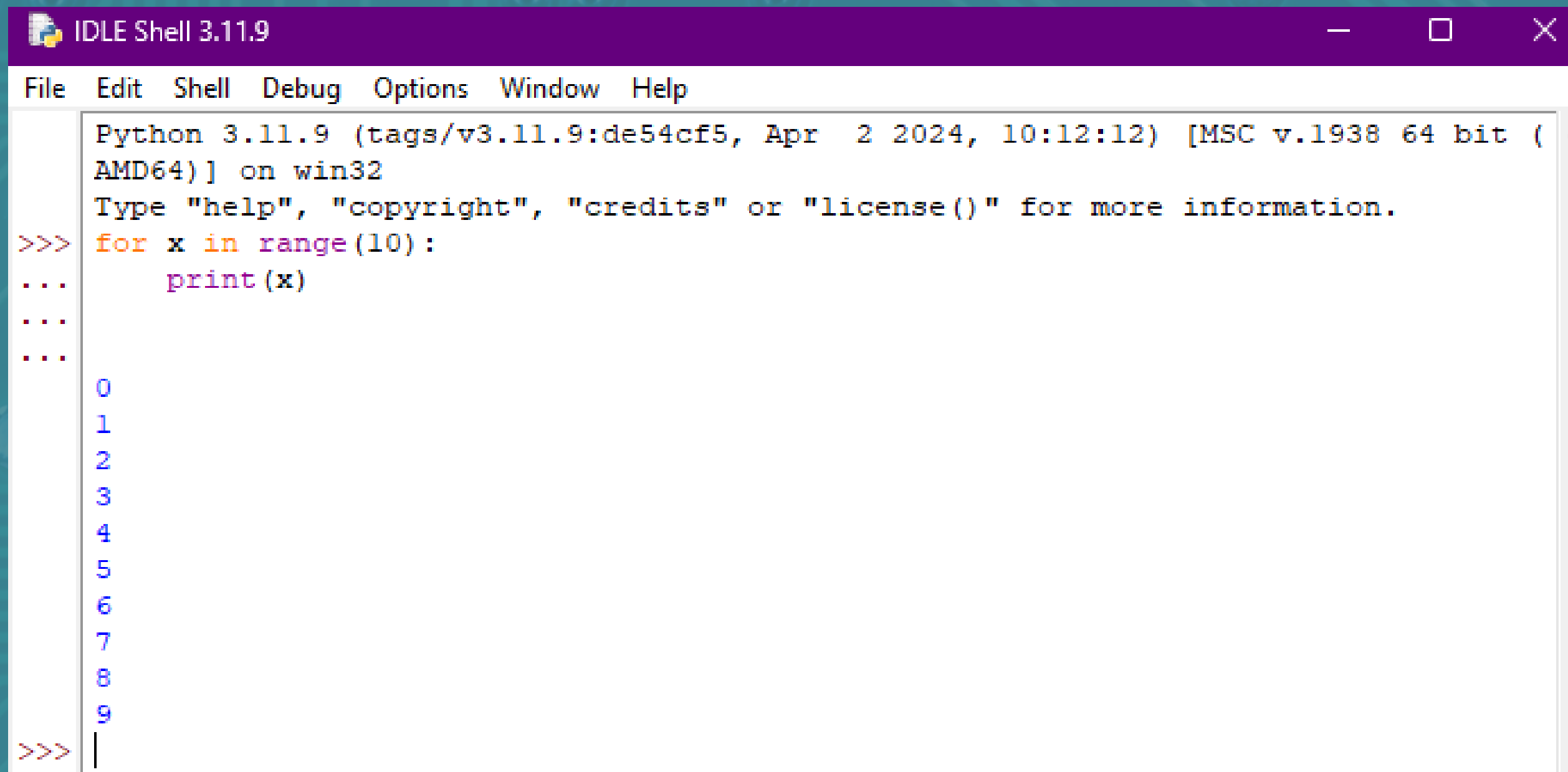
A screenshot of the IDLE Shell 3.11.9 window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following content:

```
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> age = 24
>>> if age >= 18:
...     print("$25 to eat")
... else:
...     print("Eat free!")
...
$25 to eat
>>> age = 4
>>> if age >= 18:
...     print("$25 to eat")
... else:
...     print("Eat free!")
...
Eat free!
>>> |
```

CONDITIONALS LET PROGRAMS MAKE CHOICES.

Loops

- FOR LOOP: REPEAT OVER SEQUENCE
- WHILE LOOP: REPEAT UNTIL CONDITION FALSE

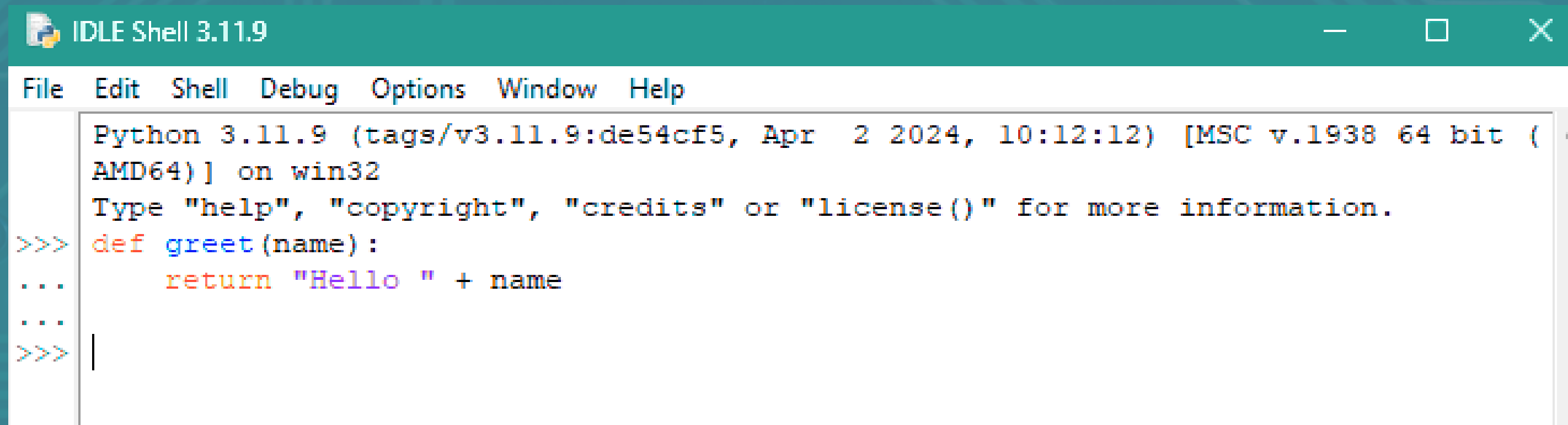
A screenshot of the IDLE Shell 3.11.9 window. The title bar is purple and says "IDLE Shell 3.11.9". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the Python 3.11.9 shell prompt and a for loop. The loop iterates from 0 to 9, printing each number on a new line. The prompt is ">>>". The output is "0", "1", "2", "3", "4", "5", "6", "7", "8", "9". The prompt is ">>>".

```
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> for x in range(10):
...     print(x)
...
...
0
1
2
3
4
5
6
7
8
9
>>> |
```

LOOPS AUTOMATE REPETITION.

Functions

- DEFINE REUSABLE BLOCKS OF CODE
- USE PARAMETERS & RETURN VALUES



```
IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> def greet(name):
...     return "Hello " + name
...
>>> |
```

FUNCTIONS KEEP CODE ORGANIZED.

List & Collections

- LISTS: ORDERED, MUTABLE
- TUPLES: ORDERED, IMMUTABLE
- DICTIONARIES: KEY-VALUE PAIRS

```
....  
>>> fruits = ["apple", "kiwi", "banana", "pineapple", "burger"]  
>>> print(fruits[2])  
banana  
>>> |
```

Side note: *The counting iteration starts at 0 and goes up to 1, 2, 3*

....

FUNCTIONS KEEP CODE ORGANIZED.

Error Handling

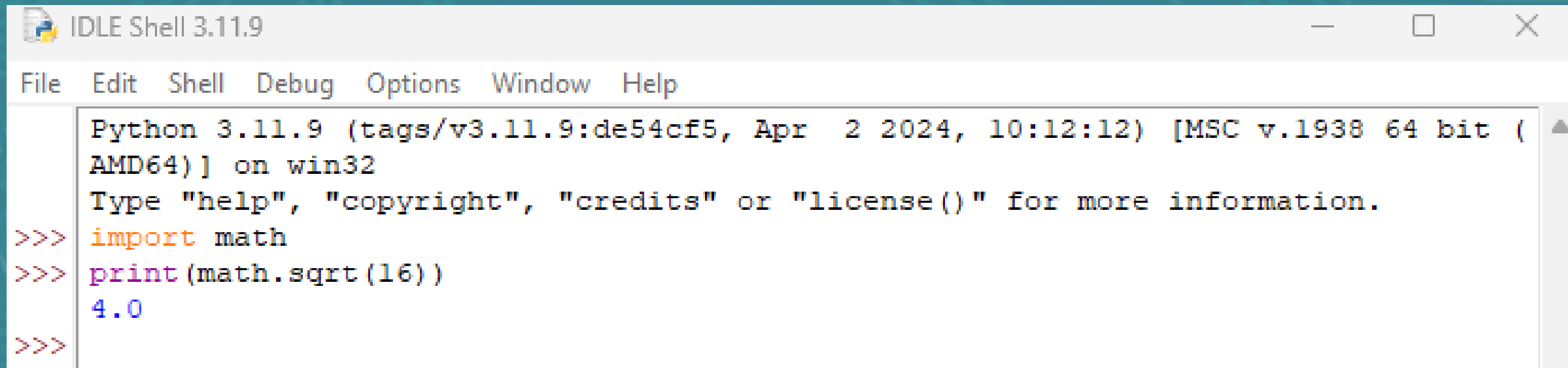
- TRY / EXCEPT BLOCKS
- PREVENT CRASHES

```
>>> try:
...     x = int("abc")
... except ValueError:
...     print("Invalid input")
...
...
Invalid input
>>> |
```

ERRORS HAPPEN — HANDLING THEM MAKES PROGRAMS ROBUST.

Modules & Libraries

- IMPORT BUILT-IN MODULES: MATH, RANDOM
- INSTALL EXTERNAL LIBRARIES: PIP INSTALL NUMPY



```
IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import math
>>> print(math.sqrt(16))
4.0
>>>
```

Getting the square root of 16 using the import math module
LIBRARIES EXTEND PYTHON'S POWER.

Mini Project

```
# CREATING A SIMPLE CALCULATOR
```

```
A = INT(INPUT("ENTER NUMBER: "))
```

```
B = INT(INPUT("ENTER NUMBER: "))
```

```
PRINT("SUM =", A + B)
```

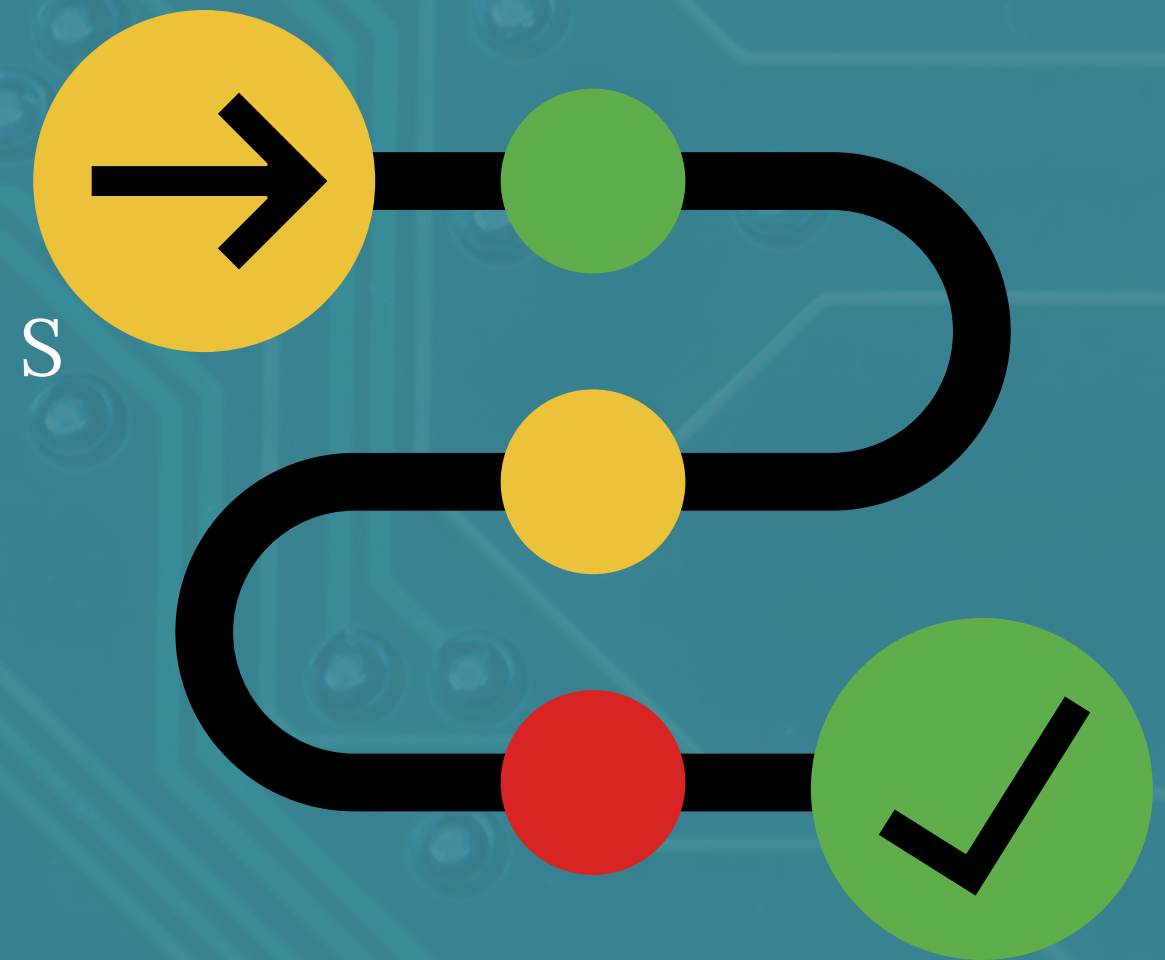
```
>>> a = int(input("Enter number: "))
Enter number: 20
>>> b = int(input("Enter number: "))
Enter number: 50
>>> print ("Sum = ", a + b)
Sum = 70
>>>
```

CHALLENGE: CAN YOU ADD TWO NAMES TOGETHER TO CREATE A FULL NAME?

LET'S BUILD A TINY PROJECT TO APPLY WHAT WE LEARNED.

Wrap-Up & Next Steps

- PYTHON BASICS: VARIABLES, LOOPS, FUNCTIONS
- PRACTICE BY CODING SMALL PROJECTS
- EXPLORE ADVANCED TOPICS: OOP, DATA SCIENCE, WEB DEV
- SIGNUP FOR A BIOGLYTCH CODING CLASS



PYTHON IS A GATEWAY TO ENDLESS POSSIBILITIES.